

Rank-Constrained Fundamental Matrix Estimation by Polynomial Global Optimization Versus the Eight-Point Algorithm

Florian Bugarin,¹ Adrien Bartoli,² Didier Henrion,^{3,4}
Jean-Bernard Lasserre,^{3,5} Jean-José Orteu,¹ Thierry Sentenac,^{1,3}

March 20, 2014

Abstract

The fundamental matrix can be estimated from point matches. The current gold standard is to bootstrap the eight-point algorithm and two-view projective bundle adjustment. The eight-point algorithm first computes a simple linear least squares solution by minimizing an algebraic cost and then projects the result to the closest rank-deficient matrix. We propose a single-step method that solves both steps of the eight-point algorithm. Using recent results from polynomial global optimization, our method finds the rank-deficient matrix that exactly minimizes the algebraic cost. In this special case, the optimization method is reduced to the resolution of very short sequences of convex linear problems which are computationally efficient and numerically stable. The current gold standard is known to be extremely effective but is nonetheless outperformed by our rank-constrained method for bootstrapping bundle adjustment. This is here demonstrated on simulated and standard real datasets. With our initialization, bundle adjustment consistently finds a better local minimum (achieves a lower reprojection error) and takes less iterations to converge.

¹Université de Toulouse; INSA, UPS, Mines Albi, ISAE; ICA (Institut Clément Ader); Campus Jarlard, F-81013 Albi, France

²Université d'Auvergne, ISIT, UMR 6284 CNRS / Université d'Auvergne, Clermont-Ferrand, France

³LAAS-CNRS, Université de Toulouse, Toulouse, France

⁴Faculty of Electrical Engineering, Czech Technical University in Prague

⁵Institut de Mathématiques de Toulouse, Université de Toulouse, Toulouse, France

Keywords : Global optimization, linear matrix inequality, fundamental matrix.

1 Introduction

The fundamental matrix has received a great interest in the computer vision community (see for instance [1, 2, 3, 4, 5, 6, 7]). This (3×3) rank-two matrix encapsulates the epipolar geometry, the projective motion between two uncalibrated perspective cameras, and serves as a basis for 3D reconstruction, motion segmentation and camera self-calibration, to name a few. Given n point matches $(\mathbf{q}_i, \mathbf{q}'_i)$, $i = 1, \dots, n$ between two images, the fundamental matrix may be estimated in two phases. The initialization phase finds some suboptimal estimate while the refinement phase iteratively minimizes an optimal but nonlinear and nonconvex criterion. The gold standard uses the *eight-point algorithm* and *projective bundle adjustment* for these two phases, respectively. A ‘good enough’ initialization is necessary to avoid local minima at the refinement phase as much as possible. The main goal of this article is to improve the current state of the art regarding the initialization phase. We here focus on input point matches that do not contain mismatches (a pair of points incorrectly associated). The problem of mismatches has been specifically addressed by the use of robust methods in the literature.

The eight-point algorithm follows two steps [1]. In its first step, it relaxes the rank-deficiency constraint and solves the following convex problem:

$$\tilde{\mathbf{F}} = \arg \min_{\mathbf{F} \in \mathbb{R}^{3 \times 3}} C(\mathbf{F}) \text{ s.t. } \|\mathbf{F}\|^2 = 1, \quad (1)$$

where C is a convex, linear least squares cost, hereinafter called the *algebraic cost*:

$$C(\mathbf{F}) = \sum_{i=1}^n \left(\mathbf{q}_i^\top \mathbf{F} \mathbf{q}_i \right)^2. \quad (2)$$

This minimization is subject to the normalization constraint $\|\mathbf{F}\|^2 = 1$. This is to avoid the trivial solution $\mathbf{F} = \mathbf{0}$. Normalization will be further discussed in section 3. The estimated matrix $\tilde{\mathbf{F}}$ is thus *not* a fundamental matrix yet. In its second step, the eight-point algorithm computes the closest rank-deficient matrix to $\tilde{\mathbf{F}}$ as:

$$\mathbf{F}_{8\text{pt}} = \arg \min_{\mathbf{F} \in \mathbb{R}^{3 \times 3}} \|\mathbf{F} - \tilde{\mathbf{F}}\|^2 \text{ s.t. } \det(\mathbf{F}) = 0. \quad (3)$$

Both steps can be easily solved. The first step is a simple linear least squares problem and the second step is solved by nullifying the least singular value of $\tilde{\mathbf{F}}$. It has been shown [4] that this simple algorithm performs extremely well in

practice, provided that the image point coordinates are *standardized* by simply rescaling them so that they lie in $[-\sqrt{2}; \sqrt{2}]^2$.

Our main contribution in this paper is an approach that solves for the fundamental matrix minimizing the algebraic cost. In other words, we find the global minimum of:

$$\tilde{\mathbf{F}} = \arg \min_{\mathbf{F} \in \mathbb{R}^{3 \times 3}} C(\mathbf{F}) \text{ s.t. } \det(\mathbf{F}) = 0 \text{ and } \|\mathbf{F}\|^2 = 1. \quad (4)$$

Perhaps more importantly, we also quantify the impact that each of $\mathbf{F}_{8\text{pt}}$ and $\tilde{\mathbf{F}}$ has when used as an initial estimate in Bundle Adjustment. Each initial estimate will lead Bundle Adjustment to its own refined estimate. The two final estimates may thus be different since, as the difference between the two initial estimates grows larger, the probability that they lie in different basins of attraction increases. Our measure quantifies:

1. how far are these two basins of attractions,
2. how many iterations will Bundle Adjustment take to converge.

The proposed algorithm uses polynomial global optimization [8, 9]. Previous attempts [10, 11, 12] in the literature differ in terms of optimization strategy and parameterization of the fundamental matrix. None solves problem (4) optimally for a general parameterization: they either do not guarantee global optimality [11, 12] or prescribe some camera configurations [10, 11, 12] (requiring typically that the epipole in the first camera does not lie at infinity). Furthermore, the main criticism made to the optimization method we use is the resolution of a hierarchy of convex linear problems of increasing size, which is computationally ineffective and numerically unstable. The proposed solution overcomes this drawback: experiments show that, in most of cases, the proposed algorithm only requires solving the second relaxation of the sequences used in the exposed optimization method.

Our experimental evaluation on simulated and real datasets compares the difference between the eight-point algorithm and ours used as initialization to bundle adjustment. We observe that (i) bundle adjustment consistently converges within less iterations with our initialization and (ii) bundle adjustment always achieves an equal or lower reprojection error with our initialization. We provide numerous examples of real image pairs from standard datasets. They all illustrate practical cases for which our initialization method allows bundle adjustment to reach a better local minimum than the eight-point algorithm.

2 State of the Art

Accurately and automatically estimating the fundamental matrix from a pair of images is a major research topic. We first review a four-class categorization of existing methods, and specifically investigate the details of existing global methods. We finally state the improvements brought by our proposed global method.

2.1 Categorizing Methods

A classification of the different methods in three categories –linear, iterative and robust– was proposed in [2]. Linear methods directly optimize a linear least squares cost. They include the eight-point algorithm [1], SVD resolution [2] and variants [3, 4, 5, 6]. Iterative methods iteratively optimize a nonlinear and nonconvex cost. They require, and are sensitive to the quality of, an initial estimate. The first group of iterative methods minimizes the distances between points and epipolar lines [13, 14]. The second group minimizes some approximation of the reprojection error [15, 16, 17, 18]. The third group of methods minimizes the reprojection error, and are equivalent to two-view projective bundle adjustment. Iterative methods typically use a nonlinear parameterization of the fundamental matrix which guarantees that the rank-deficiency constraint is met. For instance, a minimal 7-parameter update can be used over a consistent orthogonal representation [7]. Finally, robust methods estimate the fundamental matrix while classifying each point match as inlier or outlier. Robust methods use M-Estimators [19], LMedS (median least squares) [16] or RANSAC (random sampling consensus) [20]. Both LMedS and RANSAC are stochastic.

To these three categories, we propose to add a fourth one: *global methods*. Global methods attempt at finding the global minimum of a nonconvex problem. Convex relaxations have been used to combine a convex cost with the rank-deficiency constraint [11]. However, these relaxations do not converge to a global minimum and the solution’s optimality is not certified.

2.2 Global Methods

In theory, for a constrained optimization problem, global optimization methods do not require an initial guess and may be guaranteed to reach the global minimum, thereby certifying optimality. Such global methods can be separated in two classes. The methods of the first class describe the search space as exhaustively as possible in order to test as many candidate solutions as possible. Following this way, there are methods such as Monte-Carlo sampling, which test random elements satisfying constraints, and reactive tabu search [21, 22], which continues searching even after a local minimum has been found. The major drawback of

these methods is mainly in the prohibitive computation time required to have a sufficiently high probability of success. Moreover, even in case of convergence, there is no certificate of global optimality. Contrary to the methods of the first class, methods lying in the second class provide a certificate of global optimality using the mathematical theory from which they are built. Branch and Bound algorithms [23] or global optimization by interval analysis [24, 25] are some examples. However, although these methods can be faster than those of the first category, their major drawback is their lack of generality. Indeed, these methods are usually dedicated to one particular type of cost function because they use highly specific computing mechanisms to be as efficient as possible. A review of global methods may be found in [26].

A good deal of research has been conducted over the last few decades on applying global optimization methods in order to solve polynomial minimization problems under polynomial constraints. The major drawback of these applications has been the difficulty to take constraints into account. But, by solving simplified problems, these approaches have mainly been used to find a starting point for local iterative methods. However, recent results in the areas of convex and polynomial optimization have facilitated the emergence of new approaches. These have attracted great interest in the computer vision community. In particular, global polynomial optimization [8, 9] has been used in combination with a finite-epipole nonlinear parameterization of the fundamental matrix [10]. This method does not consequently cover camera setups where the epipole lies at infinity. A global convex relaxation scheme [8, 9] was used to minimize the Sampson distance [12]. Because this implies minimizing a sum of many rational functions, the generic optimization method had to be specifically adapted and lost the property of certified global optimality.

2.3 The Proposed Method

The proposed method lies in the fourth category: it is a global method. Similarly to the eight-point algorithm, it minimizes the algebraic cost, but explicitly enforces the nonlinear rank-deficiency constraint. Contrarily to previous global methods [10, 11, 12], the proposed method handles all possible camera configurations (it does not make an assumption on the epipoles being finite or infinite) and certifies global optimality. Moreover, the presented algorithm is based on the resolution of a very short sequence of convex linear problems and is therefore computationally efficient.

A large number of attempts to introduce global optimization have been made in the literature. In [11], a dedicated hierarchy of convex relaxations is defined in order to globally solve the problem of fundamental matrix estimation. The singularity constraint is taken into account by introducing additional optimization

variables, not introduced directly in the problem description. The resulting optimization algorithm is not generic and, contrary to Lasserre's hierarchy, there is no proof that the sequence of solutions of this specific hierarchy converges to the global minimum.

In [10], Lasserre's hierarchy is used jointly with the introduction of the singularity constraint in the problem description. However, the normalization constraint $\|\mathbf{F}\|^2 = 1$ is replaced by fixing, a priori, one of the \mathbf{F} coefficients to 1. Hence the other \mathbf{F} coefficients are not bounded. Consequently, there is no guarantee that the sequence of solutions $(\hat{f}_t)_{t \in \mathbb{N}}$ converges to the global minimum.

In [27] the authors minimize the Sampson distance by solving a specific hierarchy of convex relaxations built upon an epigraph formulation. This specific hierarchy is obtained by extending Lasserre's hierarchy, where the constraints are linear (LMIs), to a hierarchy of convex relaxations where constraints are polynomial (polynomial matrix inequality - PMI). Consequently, no asymptotic convergence of the hierarchy to the global optimum can be guaranteed. Moreover, even if the rank correction is achieved using the singularity constraint, the normalization is replaced by setting, a priori, the coefficient f_{33} to 1 and so the other coefficients of \mathbf{F} are not bounded.

Finally, in a very recent work [28], the algebraic error is globally minimized thanks to the resolution of seven subproblems. Each subproblem is reduced to a polynomial equation system solved via a Gröbner basis solver. The singularity constraint is satisfied thanks to the right epipole parametrization. Although this parametrization ensures that \mathbf{F} is singular while using the minimum number of parameters, this method is not practical since it would be necessary to solve 126 subproblems in order to cover all the 18 possible parameter sets [16]. Therefore it is preferable to introduce the singularity constraint directly in the problem description rather than via some parametrization of \mathbf{F} .

3 Polynomial Global Optimization

3.1 Introduction

Given a real-valued polynomial $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, we are interested in solving the problem:

$$f^* = \inf_{x \in K} f(x) \quad (5)$$

where $K \subseteq \mathbb{R}^n$ is a (not necessarily convex) compact set defined by polynomial inequalities: $g_j(x) \geq 0$, $j = 1, \dots, m$. Our optimization method is based on an

idea first described in [29]. It consists in reformulating the nonconvex global optimization problem (5) as the equivalent convex linear programming problem:

$$\hat{f} = \inf_{\mu \in \mathcal{P}(K)} \int_K f(x) d\mu, \quad (6)$$

where $\mathcal{P}(K)$ is the set of probability measures supported on K . Note that this reformulation is true for any continuous function (not necessarily polynomial) and any compact set $K \subseteq \mathbb{R}^n$. Indeed, as $f^* \leq f(x)$, then $f^* \leq \int_K f d\mu$ and thus $f^* \leq \hat{f}$. Conversely, if x^* is a global minimizer of (5), then the probability measure $\mu^* \triangleq \delta_{x^*}$ (the Dirac at x^*) is admissible for (6). Moreover, because \hat{f} is a solution of (6), the following inequality holds: $\int_K f(x) d\mu \geq \hat{f}$, $\forall \mu \in \mathcal{P}(K)$ and thus $f^* = \int_K f(x) \delta_{x^*} \geq \hat{f}$. Instead of optimizing over the finite-dimensional euclidean space K , we optimize over the infinite-dimensional set of probability measures $\mathcal{P}(K)$. Thus, Problem (6) is, in general, not easier to solve than Problem (5). However, in the special case of f being a polynomial and K being defined by polynomial inequalities, we will show how Problem (6) can be reduced to solving an (generically finite) sequence of convex linear matrix inequality (LMI) problems.

3.2 Notations and Definitions

First, given vectors $\alpha = (\alpha_1, \dots, \alpha_n)^\top \in \mathbb{N}^n$ and $x = (x_1, \dots, x_n)^\top \in \mathbb{R}^n$, we define the *monomial* x^α by:

$$x^\alpha \triangleq x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n} \quad (7)$$

and its *degree* by $\deg(x^\alpha) \triangleq \|\alpha\|_1 = \sum_{i=1}^n \alpha_i$. For $t \in \mathbb{N}$, we define \mathbb{N}_t^n the space of the n -dimensional integer vector with a norm lower than t as:

$$\mathbb{N}_t^n \triangleq \{\alpha \in \mathbb{N}^n \mid \|\alpha\|_1 \leq t\}. \quad (8)$$

Then, consider the family:

$$\{x^\alpha\}_{\alpha \in \mathbb{N}_t^n} = \{1, x_1, x_2, \dots, x_n, x_1^2, x_1 x_2, \dots, x_1 x_n, x_2 x_3, \dots, x_n^2, \dots, x_1^t, \dots, x_n^t\} \quad (9)$$

of all the monomials x^α of degree at most t , which has dimension $s(t) \triangleq \frac{(n+t)!}{t!n!}$. Those monomials form the canonical basis of the vector space $\mathbb{R}_t[x]$ of real-valued multivariate polynomials of degree at most t . Then, a polynomial $p \in \mathbb{R}_t[x]$ is understood as a linear combination of monomials of degree at most t :

$$p(x) = \sum_{\alpha \in \mathbb{N}_t^n} p_\alpha x^\alpha, \quad (10)$$

and $\mathbf{p} \triangleq (p_\alpha)_{\|\alpha\|_1 \leq t} \in \mathbb{R}^{\mathbb{N}_t^n} \simeq \mathbb{R}^{s(t)}$ is the vector of its coefficients in the monomial basis $\{x^\alpha\}_{\alpha \in \mathbb{N}_t^n}$. Its degree is equal to $\deg(p) \triangleq \max \{\|\alpha\|_1 \mid p_\alpha \neq 0\}$ and d_p denotes the smallest integer not lower than $\frac{\deg(p)}{2}$.

Example: The polynomial

$$x \in \mathbb{R}^2 \mapsto p(x) = 1 + 2x_2 + 3x_1^2 + 4x_1x_2 \quad (11)$$

has a vector of coefficients $p \in \mathbb{R}^6$ with entries $p_{00} = 1$, $p_{10} = 0$, $p_{01} = 2$, $p_{20} = 3$, $p_{11} = 4$ and $p_{02} = 0$.

Next, given $\mathbf{y} = (y_\alpha)_{\alpha \in \mathbb{N}^n} \in \mathbb{R}^{\mathbb{N}^n}$, we define the *Riesz functional* $L_{\mathbf{y}}$ by the linear form:

$$\begin{aligned} L_{\mathbf{y}} : \mathbb{R}[x] &\rightarrow \mathbb{R} \\ p = \sum_{\alpha \in \mathbb{N}^n} p_\alpha x^\alpha &\rightarrow \mathbf{y}^\top \mathbf{p} = \sum_{\alpha \in \mathbb{N}^n} p_\alpha y_\alpha. \end{aligned} \quad (12)$$

Thus, the Riesz functional can be seen as an operator that linearizes polynomials.

Example: For the polynomial (11), the Riesz functional reads

$$p(x) = 1 + 2x_2 + 3x_1^2 + 4x_1x_2 \mapsto L_{\mathbf{y}}(p) = y_{00} + 2y_{01} + 3y_{20} + 4y_{11}. \quad (13)$$

For $t \in \mathbb{N}$ and $\mathbf{y} \in \mathbb{R}^{\mathbb{N}_{2t}^n}$, the matrix $M_t(\mathbf{y})$ of size $s(t)$ defined by:

$$(M_t(\mathbf{y}))_{\alpha, \beta} = L_{\mathbf{y}}(x^\alpha x^\beta) = y_{\alpha+\beta} \quad \forall \alpha, \beta \in \mathbb{N}_t^n \quad (14)$$

is called *the moment matrix of order t of \mathbf{y}* . By construction, this matrix is symmetric and linear in \mathbf{y} . Then, given $q \in \mathbb{R}_t[x]$ and $\mathbf{q} \in \mathbb{R}^{\mathbb{N}_t^n}$ the vector of its coefficients in the monomial basis, the vector:

$$q\mathbf{y} \triangleq M_t(\mathbf{y})\mathbf{q} \in \mathbb{R}^{\mathbb{N}_t^n} \quad (15)$$

is called *the shifted vector* with respect to \mathbf{q} . $M_t(q\mathbf{y})$, the moment matrix of order t of $q\mathbf{y}$, is called *the localizing matrix of degree t of q* . This matrix is also symmetric and linear in \mathbf{y} .

Example: If $n = 2$ then:

$$\begin{aligned}
M_0(\mathbf{y}) &= y_{00}, \\
M_1(\mathbf{y}) &= \begin{pmatrix} y_{00} & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{pmatrix}, \\
M_2(\mathbf{y}) &= \begin{pmatrix} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{pmatrix},
\end{aligned} \tag{16}$$

and if $q(x) = a + 2x_1^2 + 3x_2^2$ then:

$$M_1(q\mathbf{y}) = \begin{pmatrix} ay_{00} + 2y_{20} + 3y_{02} & ay_{10} + 2y_{30} + 3y_{12} & ay_{01} + 2y_{21} + 3y_{03} \\ ay_{10} + 2y_{30} + 3y_{12} & ay_{20} + 2y_{40} + 3y_{22} & ay_{11} + 2y_{31} + 3y_{13} \\ ay_{01} + 2y_{21} + 3y_{03} & ay_{11} + 2y_{31} + 3y_{13} & ay_{02} + 2y_{22} + 3y_{04} \end{pmatrix}. \tag{17}$$

Finally, recall that a symmetric matrix $F \in \mathbb{S}^n$ is positive semidefinite, denoted by $F \succeq 0$, if and only if $x^\top F x \geq 0$, $\forall x \in \mathbb{R}^n$ or equivalently, if and only if the minimum eigenvalue of F is non-negative. A *linear matrix inequality (LMI)* is a convex constraint:

$$F_0 + \sum_{k=1}^n x_k F_k \succeq 0, \tag{18}$$

on a vector $x \in \mathbb{R}^n$, where matrices $F_k \in \mathbb{S}^m$, $k = 0, \dots, n$ are given.

3.3 Optimization Method

Let f be a real-valued multivariate polynomial, Problem (6) can be reduced to a convex linear programming problem. Indeed, if $f(x) = \sum_{\alpha \in \mathbb{N}^n} f_\alpha x^\alpha$ then:

$$\int_K f d\mu = \int_K \sum_{\alpha \in \mathbb{N}^n} f_\alpha x^\alpha d\mu = \sum_{\alpha \in \mathbb{N}^n} f_\alpha \int_K x^\alpha d\mu = L_{\mathbf{y}}(f) \tag{19}$$

where each coordinate y_α of the infinite sequence $\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n}$ is equal to $\int_K x^\alpha \mu(dx)$, also called *the moment of order α* . Consequently, if f is polynomial, then Problem (6) is equivalent to:

$$\begin{aligned}
\hat{f} &= \inf L_{\mathbf{y}}(f) \\
\text{s.t. } &y_0 = 1 \\
&\mathbf{y} \in \mathcal{M}_K.
\end{aligned} \tag{20}$$

with:

$$\mathcal{M}_K \triangleq \left\{ \mathbf{y} \in \mathbb{R}^{\mathbb{N}^n} \mid \exists \mu \in \mathcal{M}_+(K) \text{ such that } y_\alpha = \int_K x^\alpha d\mu \quad \forall \alpha \in \mathbb{N}^n \right\} \quad (21)$$

and $\mathcal{M}_+(K)$ is the space of finite Borel measures supported on K . Remark that the constraint $y_0 = 1$ is added in order to impose that if $\mathbf{y} \in \mathcal{M}_K$ then \mathbf{y} represents a measure in $\mathcal{P}(K)$ (and no longer in $\mathcal{M}_+(K)$). Although Problem (20) is a convex linear programming problem, it is difficult to describe the convex cone \mathcal{M}_K with simple constraints on \mathbf{y} . But, the problem “ $\mathbf{y} \in \mathcal{M}_K$ ”, also called *K-moment problem*, is solved when K is a *basic semi-algebraic set*, namely:

$$K \triangleq \{x \in \mathbb{R}^n \mid g_1(x) \geq 0, \dots, g_m(x) \geq 0\} \quad (22)$$

where $g_j \in \mathbb{R}[x]$, $\forall j = 1, \dots, m$. Note that K is assumed to be compact. Then, without loss of generality, we assume that one of the polynomial inequalities $g_j(x) \geq 0$ is of the form $R^2 - \|x\|_2^2 \geq 0$ where R is a sufficiently large positive constant. In this particular case, \mathcal{M}_K can be modelled using LMI conditions on the matrices $M_t(\mathbf{y})$ and $M_t(g_j\mathbf{y})$, $j = 1, \dots, m$. More precisely, thanks to the Putinar Theorem [30, 31], we have:

$$\mathcal{M}_K = \mathcal{M}_{\succeq}(g_1, \dots, g_m), \quad (23)$$

where:

$$\mathcal{M}_{\succeq}(g_1, \dots, g_m) \triangleq \left\{ \mathbf{y} \in \mathbb{R}^{\mathbb{N}^n} \mid M_t(\mathbf{y}) \succeq 0, M_t(g_j\mathbf{y}) \succeq 0 \quad \forall j = 1, \dots, m \quad \forall t \in \mathbb{N} \right\}. \quad (24)$$

Then, Problem (6) is equivalent to:

$$\begin{aligned} \hat{f} = & \inf_{\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n}} L_{\mathbf{y}}(f) \\ \text{s.t. } & y_0 = 1 \\ & M_t(\mathbf{y}) \succeq 0 \\ & M_t(g_j\mathbf{y}) \succeq 0 \quad j = 1, \dots, m \quad \forall t \in \mathbb{N}. \end{aligned} \quad (25)$$

To summarize, if f is polynomial and K a semi-algebraic set, then Problem (5) is equivalent to a convex linear programming problem with an infinite number of linear constraints on an infinite number of decision variables. Now, for $t \geq d_K \triangleq \max(d_f, d_{g_1}, \dots, d_{g_m})$ consider the finite-dimensional truncations of Problem (25):

$$\mathcal{Q}_t \triangleq \left\{ \begin{aligned} \hat{f}_t & \triangleq \min_{\mathbf{y} \in \mathbb{R}^{\mathbb{N}_{2t}^n}} L_{\mathbf{y}}(f) \\ \text{s.t. } & y_0 = 1 \\ & M_t(\mathbf{y}) \succeq 0, \\ & M_{t-d_{g_j}}(g_j\mathbf{y}) \succeq 0 \quad \forall j \in \{1, \dots, m\}. \end{aligned} \right. \quad (26)$$

By construction, \mathcal{Q}_t , $t \in \mathbb{N}$ generates a hierarchy of LMI relaxations of Problem (25) [8], where each \mathcal{Q}_t , $t \in \mathbb{N}$, is concerned with moment and localizing matrices of fixed size t . Each relaxation (26) can be solved by using public-domain implementations of primal-dual interior point algorithms for semidefinite programming (SDP) [32, 33, 34, 35, 36]. When *the relaxation order* $t \in \mathbb{N}$ tends to infinity, we obtain the following results [8, 37]:

$$\hat{f}_t \leq \hat{f}_{t+1} \leq \hat{f} \text{ and } \lim_{t \rightarrow +\infty} \hat{f}_t = \hat{f}. \quad (27)$$

Practice reveals that this convergence is fast and very often finite, i.e. there exists a finite t_0 such that $\hat{f}_t = \hat{f}$, $\forall t \geq t_0$. In fact, finite convergence is guaranteed in a number of cases (e.g. discrete optimization) and very recent results by Nie [37] show that finite convergence is even generic as well as an optimal solution \mathbf{y}_t^* of (26).

Example: Consider the polynomial optimization problem

$$\begin{aligned} \hat{f} = \min_{x \in \mathbb{R}^2} \quad & -x_2 \\ \text{s.t.} \quad & 3 - 2x_2 - x_1^2 - x_2^2 \geq 0 \\ & -x_1 - x_2 - x_1x_2 \geq 0 \\ & 1 + x_1x_2 \geq 0. \end{aligned} \quad (28)$$

The first LMI relaxation \mathcal{Q}_1 is

$$\begin{aligned} \hat{f}_1 = \min_{\mathbf{y} \in \mathbb{R}^6} \quad & -y_{01} \\ \text{s.t.} \quad & y_{00} = 1 \\ & \begin{pmatrix} y_{00} & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{pmatrix} \succeq 0 \\ & 3y_{00} - 2y_{01} - y_{20} - y_{02} \geq 0 \\ & -y_{10} - y_{01} - y_{11} \geq 0 \\ & y_{00} + y_{11} \geq 0, \end{aligned} \quad (29)$$

and the second LMI relaxation \mathcal{Q}_2 is

$$\begin{aligned}
\hat{f}_2 = \min_{\mathbf{y} \in \mathbb{R}^{15}} \quad & -y_{01} \\
\text{s.t.} \quad & y_{00} = 1 \\
& \begin{pmatrix} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{pmatrix} \succeq 0, \\
& \begin{pmatrix} 3y_{00} - 2y_{01} - y_{20} - y_{02} & 3y_{10} - 2y_{11} - y_{30} - y_{12} & 3y_{01} - 2y_{02} - y_{21} - y_{03} \\ 3y_{10} - 2y_{11} - y_{30} - y_{12} & 3y_{20} - 2y_{21} - y_{40} - y_{22} & 3y_{11} - 2y_{12} - y_{31} - y_{13} \\ 3y_{01} - 2y_{02} - y_{21} - y_{03} & 3y_{11} - 2y_{12} - y_{31} - y_{13} & 3y_{02} - 2y_{03} - y_{22} - y_{04} \end{pmatrix} \succeq 0 \\
& \begin{pmatrix} -y_{10} - y_{01} - y_{11} & -y_{20} - y_{11} - y_{21} & -y_{11} - y_{02} - y_{12} \\ -y_{20} - y_{11} - y_{21} & -y_{30} - y_{21} - y_{31} & -y_{21} - y_{31} - y_{21} \\ -y_{11} - y_{02} - y_{12} & -y_{21} - y_{12} - y_{22} & -y_{12} - y_{03} - y_{13} \end{pmatrix} \succeq 0 \\
& \begin{pmatrix} y_{00} + y_{11} & y_{10} + y_{21} & y_{01} + y_{12} \\ y_{10} + y_{21} & y_{20} + y_{31} & y_{11} + y_{22} \\ y_{01} + y_{12} & y_{11} + y_{22} & y_{02} + y_{13} \end{pmatrix} \succeq 0.
\end{aligned} \tag{30}$$

It can be checked that $\hat{f}_1 = -2 \leq \hat{f}_2 = \hat{f} = -\frac{1+\sqrt{5}}{2}$. Note that the constraint $3 - 2x_2 - x_1^2 - x_2^2 \geq 0$ certifies boundedness of the feasibility set.

However, we do not know a priori at which relaxation order t_0 the convergence occurs. Practically, to detect whether the optimal value is attained, we can use conditions on the rank of the moment and localization matrices. Indeed, let $\mathbf{y}^* \in \mathbb{R}^{\mathbb{N}_d^n}$ be a solution of Problem (26) at a given relaxation order $t \geq d_K$, if:

$$\text{rank}(M_t(\mathbf{y}^*)) = \text{rank}(M_{t-d_K}(\mathbf{y}^*)) \tag{31}$$

then $\hat{f}_t = \hat{f}$. In particular, if $\text{rank}(M_t(\mathbf{y}^*)) = 1$ then condition (31) is satisfied. Moreover, if these rank conditions are satisfied, then we can use numerical linear algebra to extract $\text{rank}(M_t(\mathbf{y}^*))$ global optima for Problem (5). We do not describe the algorithm in this article, but the reader can refer to [30, Sections 4.3] for more advanced information. Figure 3.3 summarizes the optimization process.

A Matlab interface called GloptiPoly [38] has been designed to construct Lasserre's LMI relaxations in a format understandable by any SDP solver interfaced via YALMIP [39]. It can be used to construct an LMI relaxation (26) of a given order corresponding to a polynomial optimization problem (5) with given polynomial data entered symbolically. A numerical algorithm is implemented in GloptiPoly to detect global optimality of an LMI relaxation, using the rank

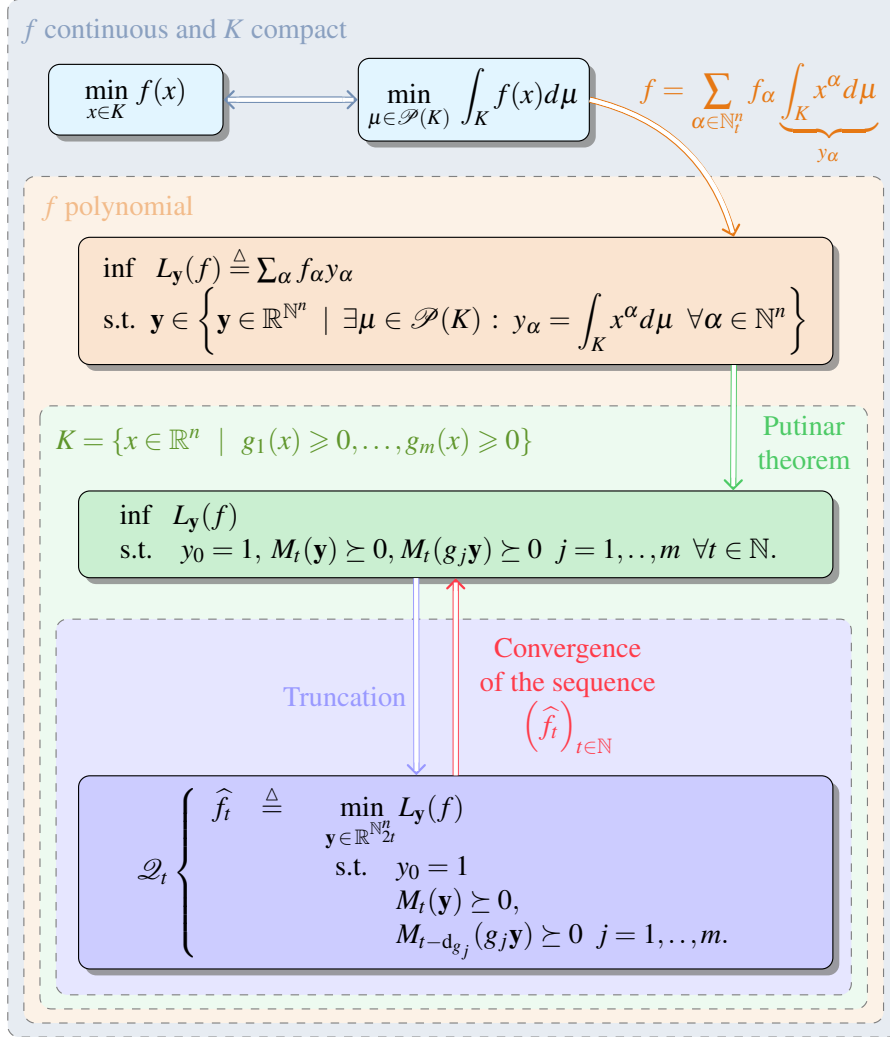


Figure 1: Polynomial optimization process; see the main text for details.

tests (31). The algorithm also extracts numerically the global optima from the moment matrix. This approach has been successfully applied to globally solve various polynomial optimization problems (see [30] for an overview of results and applications). In computer vision this approach was first introduced in [27] and used in [12].

3.4 Application to Fundamental Matrix Estimation

An important feature of our approach is that the singularity constraint can be directly satisfied by our solution and that we do *not* need an initial estimate as in other methods. Nevertheless, the problem being homogeneous, an additional normalization constraint is needed to avoid the trivial solution $\mathbf{F} = \mathbf{0}$. This is generally done by setting one of the coefficients of the \mathbf{F} matrix to 1. However with such a normalization there is no guarantee that the feasible set is compact which is a necessary condition for the convergence of our polynomial optimization method. Moreover, this normalisation excludes a priori some geometric configurations. Thus, to guarantee compactness of the feasible set and to avoid the trivial solution, we include the additional normalization constraint $\|\mathbf{F}\|^2 = 1$.

Alternatively, the singularity constraint can be inferred by parameterizing the \mathbf{F} matrix using one or two epipoles. For instance, in [10] the authors have tried to use polynomial optimization to estimate the \mathbf{F} matrix, parametrized by using a single epipole. However such an approach has several drawbacks. Firstly, the coefficients of \mathbf{F} are not bounded and the convergence of the method is not guaranteed. Secondly, using the epipole explicitly increases the degree of the polynomial criterion and consequently, the size of the corresponding relaxations in the hierarchy. This results in a significant increase of the computational time. Finally, the choice of this parametrization is arbitrary and does not cover all camera configurations.

Our method is summarized in Algorithm 1 below. Its main features are:

- In contrast with [27, 12] the optimization problem is formulated with an *explicit* Frobenius norm constraint on the decision variables. This enforces compactness of the feasibility set which is included in the Euclidean ball of radius 1. We have observed that enforcing this Frobenius norm constraint has a dramatic influence on the overall numerical behavior of the SDP solver, especially with respect to convergence and extraction of global minimizers.
- We have chosen the SDPT3 solver [40, 35] since our experiments revealed that for our problem it was the most efficient and reliable solver.

- We force the interior-point algorithm to increase the accuracy as much as possible, overruling the default parameter set in SDPT3. Then the solver runs as long as it can make progress.
- Generically, polynomial optimization problems on a compact set have a unique global minimizer; this is confirmed in our numerical experiments as the moment matrix has almost always rank-one (which certifies global optimality) at the second SDP relaxation of the hierarchy. In some (very few) cases, due to the numerical extraction, the global minimum is not fully accurate but yet largely satisfactory.

Algorithm 1 Polynomial optimization for fundamental matrix estimation

Require: Matched points $(\mathbf{q}_i, \mathbf{q}'_i), i = 1, \dots, n$

- 1: Create the cost function $Crit = \sum_{i=1}^n (\mathbf{q}'_i{}^\top \mathbf{F} \mathbf{q}_i)^2$:

```

mpol('F',3,3);
for k = 1:size(q1)
    n(k) = (q2'*F*q1)^2;
end;
Crit = sum(n);

```
 - 2: Create the constraints $\det(\mathbf{F}) = 0$ and $\|\mathbf{F}\|^2 = 1$:

```

K_det = det(F) == 0; K_fro = trace(F*F') == 1;

```
 - 3: Fix the accuracy of the solver to 0, then the solver runs as long as it can make progress:

```

pars.eps = 0; mset(pars);

```
 - 4: Change the default SDP solver to SDPT3:

```

mset('yalmip',true); mset(sdpsettings('solver','sdpt3'));

```
 - 5: Form the second LMI relaxation of the problem:

```

P = msdp(min(crit),K_det,K_fro,2);

```
 - 6: Solve the second LMI relaxation:

```

msol(P);

```
-

4 Experimental Results

This section presents results obtained by the test procedure presented below with the 8-point method and our global method. First, criteria to evaluate the performance of a fundamental matrix estimate are described. Next, the evaluation methodology is detailed. Experiments were then carried out on synthetic data to

test the sensitivity to noise and to the number of point matches. Finally, experiments on real data were performed to confirm previous results and to study the influence of the type of motion between the two images.

4.1 Evaluation Criteria

Various evaluation criteria were proposed in the literature [16] to evaluate the quality of a fundamental matrix estimate. Driven by practice, a fundamental matrix estimate \mathbf{F} is evaluated with respect to the behavior of its subsequent refinement by projective bundle adjustment. Bundle Adjustment from two uncalibrated views is described as a minimization problem. The cost function is the RMS reprojection errors. The unknowns are the 3D points \mathbf{Q}_i , $i = 1, \dots, n$ and the projection matrices \mathbf{P} and \mathbf{P}' . The criteria we use are:

1. The initial reprojection error written $e_{\text{Init}}(\mathbf{F})$.
2. The final reprojection error $e_{\text{BA}}(\mathbf{F})$.
3. The number of iterations taken by Bundle Adjustment to converge, $\text{Iter}(\mathbf{F})$.

These three criteria assess whether the estimates provided by the two methods, denoted by \mathbf{F}_{8pt} and \mathbf{F}_{Gp} , are in a ‘good’ basin of attraction. Indeed, the number of iterations gives an indication of the distance between the estimate and the optimum while $e_{\text{BA}}(\mathbf{F})$ gives an indication on the quality of the optimum.

4.2 Evaluation Method

The following algorithm summarizes our evaluation method:

Evaluate(\mathbf{F})

1. **Inputs:** fundamental matrix estimate \mathbf{F} , n point matches $(\mathbf{q}_i, \mathbf{q}'_i)$, $i = 1, \dots, n$
2. Form initial projective cameras [41]:

Find the second epipole from $\mathbf{F}^\top \mathbf{e}' \sim \mathbf{0}_{(3 \times 1)}$

Find the canonical plane homography $\mathbf{H}^* \sim [\mathbf{e}]_\times \mathbf{F}$

Set $\mathbf{P} \sim [\mathbf{I}_{(3 \times 3)} \ \mathbf{0}_{(3 \times 1)}]$ and $\mathbf{P}' \sim [\mathbf{H}^* \ \mathbf{e}']$

3. Form initial 3D points [42]:

Triangulate each point independently by minimizing the reprojection error

4. Compute $e_{\text{Init}}(\mathbf{F})$
5. Run two-view uncalibrated bundle adjustment [19]
6. Compute $e_{\text{BA}}(\mathbf{F})$ and $\text{Iter}(\mathbf{F})$
7. **Outputs:** $e_{\text{Init}}(\mathbf{F})$, $e_{\text{BA}}(\mathbf{F})$ and $\text{Iter}(\mathbf{F})$

4.3 Experiments on Simulated Data

4.3.1 Simulation Procedure

For each simulation series and for each parameter of interest (noise, number of points and number of motions), the same methodology is applied with the following four steps:

1. For two given motions between two successive images ($[\mathbf{R}_k \ t_k]$) and for a given matrix \mathbf{K} of internal parameters, a set of 3D points $(\mathbf{Q}_i)_i$, $i = 1, \dots, n$ is generated and two projection matrices \mathbf{P} and \mathbf{P}' are defined. In practice, the rotations matrices, \mathbf{R}_1 and \mathbf{R}_2 , of two motions are defined by:

$$\mathbf{R}_k \triangleq \begin{bmatrix} \cos(\theta_k) & 0 & \sin(\theta_k) \\ 0 & 1 & 0 \\ -\sin(\theta_k) & 0 & \cos(\theta_k) \end{bmatrix} \text{ with } \begin{cases} \theta_1 = \frac{\pi}{3} \\ \text{and} \\ \theta_2 = \frac{\pi}{6} \end{cases} \quad (32)$$

and their translation vectors by $t_1 = (20, 0, 5)^\top$ and $t_2 = (6, 0, 0)^\top$. These matrices are chosen such that $[\mathbf{R}_1, t_1]$ is a large movement and $[\mathbf{R}_2, t_2]$ is a small movement (see Figure 2). We simulated points lying in a cube with 10 meter side length. The first camera looks at the center of the cube and it is located 15 meters from the center of the cube. The focal length of the camera is 700 pixels and the resolution is 640×480 pixels.

2. Thanks to projection matrices $\mathbf{P} = \mathbf{K}[\mathbf{R}_1, \mathbf{t}_1]$ and $\mathbf{P}' = \mathbf{K}[\mathbf{R}_2, \mathbf{t}_2]$, the set of 3D points $(\mathbf{Q}_i)_i$ is projected into the two images as $(\mathbf{q}_i, \mathbf{q}'_i)_i$. At each of their pixel coordinates, a centered Gaussian noise with a variance σ^2 is added. In order to have statistical evidence, the results are averaged over 100 trials.
3. The resulting noisy points $(\tilde{\mathbf{q}}_i, \tilde{\mathbf{q}}'_i)_i$ are used to estimate \mathbf{F} by our method \mathbf{F}_{Gp} and the reference 8-point method \mathbf{F}_{8pt} .
4. Finally, via our evaluation procedure we evaluate the estimation error with respect to the noise standard deviation σ and the number of points n .

4.3.2 Sensitivity to Noise

We tested in two simulation series the influence of σ ranging from 0 to 2 pixels. The number of simulated points is 50. The first (resp. second) simulation series is based on the first motion $[\mathbf{R}_1 \ t_1]$ (resp. the second motion $[\mathbf{R}_2 \ t_2]$). Figure 3 gathers the influence of noise on the evaluation criteria. The first line shows the reproduction errors before, $e_{\text{Init}}(\mathbf{F})$, and after $e_{\text{BA}}(\mathbf{F})$ refinement through Bundle Adjustment with respect to the noise standard deviation. The second line shows the number of iterations $\text{Iter}(\mathbf{F})$ of the Bundle Adjustment versus the noise standard deviation. The first (resp. second) row concerns the first (resp. second) motion.

For the two motions, re-projection errors, $e_{\text{Init}}(\mathbf{F})$ or $e_{\text{BA}}(\mathbf{F})$, increase with the same slope when the noise level increases. Notice that for both movements, the Bundle-Adjustment step does not improve the results. Indeed, the noise gaussian noise is added to the projections $(\mathbf{q}_i, \mathbf{q}'_i)_i$. So this is noise which in practice would be produced by the extraction points process. Thus the solution produced by the resolution of the linear system is very close to the optimum and does not need to be refined. The initial solution provided by the triangulation step is then very close to a local minimum of the Bundle Adjustment problem. Moreover, the variation of the errors of initial re-projection before ($8pt - \text{Init}$ and $Gp - \text{Init}$) and after ($8pt - \text{BA}$ and $Gp - \text{BA}$) Bundle Adjustment versus the noise standard deviation is linear. However, the number of iterations needed for convergence is different in the two methods. The initial estimate of the triangulation computed from \mathbf{F}_{Gp} is closer to the local minimum than that obtained from \mathbf{F}_{8pt} . For the first motion (large displacement between camera 1 and 2), the number of iterations of the global method (in green) remains smaller than for the 8-point method (in blue) even though their difference seems to decrease when the noise level is high ($\sigma > 1$). For a significant displacement the quality of the estimate \mathbf{F} by the global method remains better even though the difference in quality diminishes with the noise level. Conversely, for the second motion (small displacement between the camera 1 and 2) both methods are equivalent since the difference in quality is only significant for a high level of noise ($\sigma > 1$). This is logical as the movement is less important. As a conclusion, the 8-point method provides a solution equivalent to that obtained with the global method when the displacement is not too important. For more significant movements the provided solution is not so close even though still in the same basin of attraction of a local minimum.

4.3.3 Influence of the Number of Points

In this experiment, we kept the noise level constant with a standard deviation $\sigma^2 = 0.5$ pixels. We tested the influence of the number of matches $(\mathbf{q}_i, \mathbf{q}'_i)_i$ on the

quality of the resulting estimate of \mathbf{F} . The number of points N varied from 10 to 100. Two simulation series are also carried out with the two motions.

Figure 4 brings with the same organization the evaluation criteria. It displays the influence of the number of matches for estimating \mathbf{F} on the re-projection errors and on the number of iterations. For both motions and for a sufficiently high number of matches ($N > 50$), re-projection errors, before and after refinement with Bundle Adjustment, or the number of iterations versus the number of matches converge to the same asymptote. From a high number of matches, the initial estimate from triangulation computed with \mathbf{F}_{8pt} and with \mathbf{F}_{Gp} are both in the same basin of attraction for the Bundle Adjustment problem. However, for a number of matches smaller than 50, the number of iterations to converge is smaller for given re-projection errors. The quality of the estimation by the global method seems better. The initial estimate from triangulation computed with \mathbf{F}_{8pt} goes away from the basin of convergence whereas the one computed with \mathbf{F}_{Gp} remains in the basin.

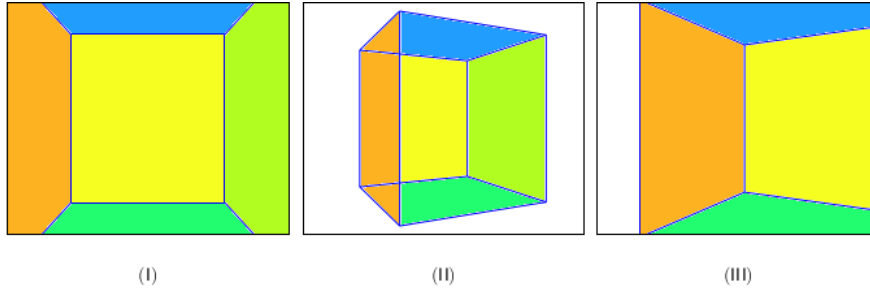


Figure 2: Projection of the cube in the camera on initial position (I) and in the camera after applying the rigid transformation $[\mathbf{R}_1 \ t_1]$ (II) and the rigid transformation $[\mathbf{R}_2 \ t_2]$ (III).

4.3.4 Influence of the Number of Points with Wide Baseline

In order to sustain the previous behavior, for a noise standard deviation of $\sigma^2 = 1$ pixel and for the significant displacement $[\mathbf{R}_1 \ t_1]$, the influence of the number of matching points on the re-projection errors and on the number of iterations was tested. In this difficult context, Figure 5 demonstrates that the initial estimate computed from \mathbf{F}_{Gp} is always closer to the local minimum than that computed from \mathbf{F}_{8pt} . No matter what is the number of matching points, the number of iterations needed to converge is always smaller.

As a conclusion, the quality of solutions obtained by both methods is almost identical when the movement is not too important, the number of matching points is sufficiently large, and the noise level is not too high. However, when one of

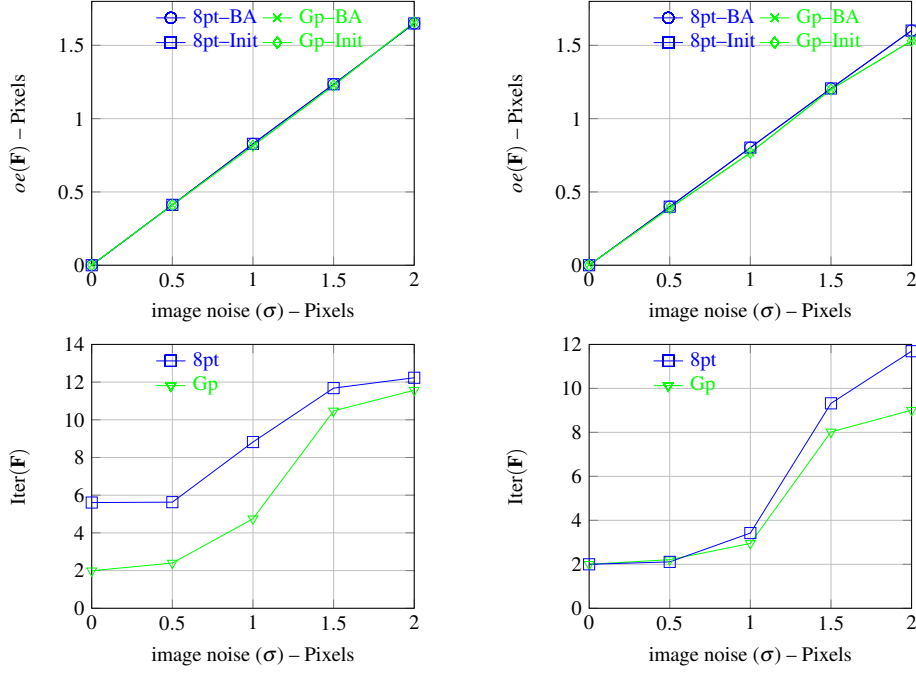


Figure 3: For two movements, $[\mathbf{R}_1 t_1]$ (left column) and $[\mathbf{R}_2 t_2]$ (right column), reprojection errors and number of iterations measured against image noise.

these three parameters varies then the 8-point method lacks precision whereas the global method still allows Bundle Adjustment to converge to the global minimum. The 8-point method computes the projection of an unconstrained local minimizer on the feasible set whereas the global method provides a global minimizer of the constrained optimization problem. It is already surprising that even for good values of the three parameters the resulting solutions are not too far apart. But for worst values of the parameters it would be even more surprising.

4.4 Experiments on Real Data

The evaluation criteria remain the same, $e_{\text{Init}}(\mathbf{F})$, $e_{\text{BA}}(\mathbf{F})$ and $\text{Iter}(\mathbf{F})$ and the computation time is added. Two experiments were carried out with two sets of images that illustrate different motions between two successive images.

4.4.1 Experiment 1

The first set of four images (see Table 6) shows all possible epipolar configurations (right or left epipole at infinity ...). With four images, six motions between a pair of images are possible: A – B, A – C, A – D, B – C, B – D and C – D. For

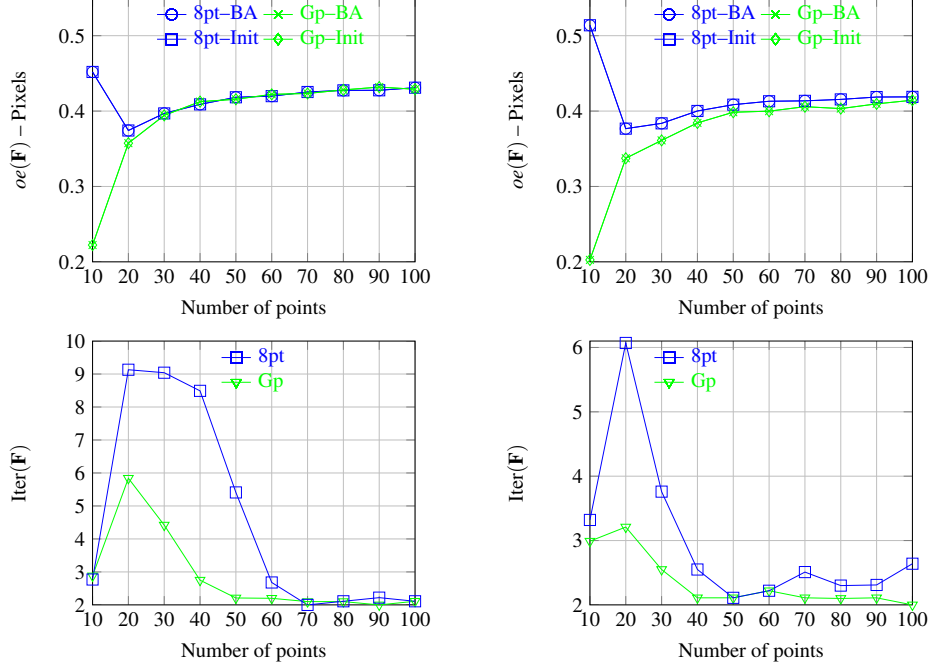


Figure 4: For two movements, $[\mathbf{R}_1 t_1]$ (left column) and $[\mathbf{R}_2 t_2]$ (right column), reprojection errors and number of iterations measured against number of points for a gaussian noise with a variance fixed to 0.5.

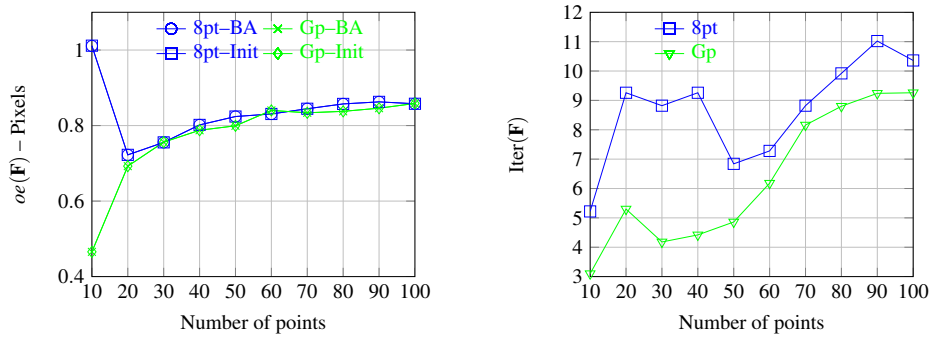


Figure 5: For the movement $[\mathbf{R}_1 t_1]$, reprojection errors and number of iterations measured against number of points for a gaussian noise with a variance fixed to 1 (left and right).

every pair of images, 60 matches are available to compute an estimate of \mathbf{F} . The values of the evaluation criteria are summarized in table 6. No matter what pair of images is used, the re-projection errors and the number of iterations are almost always better when \mathbf{F}_{Gp} is used as initial guess. In addition, for three motions (A – C, A – D and C – D), in contrast with the initial guess \mathbf{F}_{Gp} , the initial guess from the 8-point method is not in a better basin of attraction. This may explain why the initial re-projection errors $e_{\text{Init}}(\mathbf{F})$ are sometimes larger for \mathbf{F}_{Gp} as the initial guess may be in a good basin of attraction but with a larger re-projection error. For the four motions A – B, B – C, B – D and B – D, both initializations are in the same basin of attraction but the number of iterations demonstrates that the initial guess from the global method is always closer to the local minimizer. Finally, even though the computation time of the latter is significantly larger than for the 8-point method, it still remains compatible with a practical use.

4.4.2 Experiment 2

The second experiment compares the two methods on large motions. It is based on many series of images. First we test our algorithm with the classic series *Library*, *Merton*, *dinosaur* and *house* that are available at www.robots.ox.ac.uk/~vgg/data/data-mview.html. For the set of three images of *Library* and *Merton* serie, Table 7, 8, 9 and Table 10 demonstrate that the quality of the solution achieved by the global method is always better than with the 8-point method (in some cases both solutions are very close).

We also conducted the same tests on other pairs of images. For the first pair, we used images from a standard cylinder graciously provided by the company NOOME0 (<http://www.noomeo.eu/>). This cylinder is use to evaluate the accuracy of 3D reconstructions. Matched points are calculated with digital image correlation method. They are located in a window inside the cylinder. Thus, we have 6609 pairs $(\mathbf{q}_i, \mathbf{q}_i)_i$ matched to sub-pixel precision. Results are presented in the Table 11. We observe that the computation time of the 8-point method exceeds one second. This is due to the large number of matched points which leads to the resolution of a large linear system. However, as the points are precisely matched, this system is well conditioned. But the quality of the fundamental matrix estimated with the 8-point method is not sufficient to properly initialize the Bundle-Adjustment because the final re-projection error is 1.47 pixels. At the same time, even if the number of iterations is larger, our global method supplies a good estimation because the final re-projection error is 0.25 pixels. Furthermore, the calculation time remains constant in approximately 2 seconds. For the second pair, we use images taken by an endoscope. Table 12 shows the results obtained on this difficult case. As for the previous example, we observe that the fundamental matrix estimated by our global method is good quality because the final

error is 0.93 pixels. At the same time, Bundle Adjustment puts more iterations to converge on a less precise solution when we use \mathbf{F}_{8pt} to initialize it.

For the set of 36 images of the *Dinosaur* series and 9 images of the *house* series, we tested the influence of motion amplitude between a pair of image on the quality of the resulting estimates obtained by both methods. For this purpose, we had both estimates with all possible motions $((0, 1), (1, 2), (2, 3), \dots)$ with 1-image distance, then all possible motions $((0, 2), (1, 3), (2, 4), \dots)$ with 2-image distance, and so on. With this process, we can measure the influence of the average angle on the quality of the fundamental matrix estimated by both methods. Figures from 13 to 18 shows the average of re-projection errors and the average of number of iterations with respect to average angle for the two series. The re-projection error after Bundle Adjustment is always smaller with the global method and with always a smaller number of iterations. Next, the larger the movement the more the solution by both methods deteriorates. But the deterioration is larger for the 8-point method than for the global method. One may also observe that in some cases the re-projection error before Bundle Adjustment is in favor of the 8-point method. In analogy with the real cases studied before, this may be due to the fact that for these cases the initial guess \mathbf{F}_{Gp} is in a basin of attraction with a better local minimum than in the basin of attraction associated with \mathbf{F}_{8pt} , but the ‘distance’ between the initial guess and the corresponding local minimizer is larger for \mathbf{F}_{Gp} than for \mathbf{F}_{8pt} . Indeed in such cases the number of iterations is larger for \mathbf{F}_{Gp} than for \mathbf{F}_{8pt} .

5 Conclusion

We have studied the problem of estimating globally the fundamental matrix over nine parameters and under rank and normalisation constraints. We have proposed a polynomial-based approach which enables one to estimate the fundamental matrix with good precision. More generally, we have shown how to modify the constraints on the numerical certificate of optimality to obtain fast and robust convergence. The method converges in a reasonable amount of time compared to other global optimization methods.

From computational experiments conducted on both simulated and real data we conclude that the global method always provides an accurate initial estimation for the subsequent bundle adjustment step. Moreover, we have shown that if the eight-point method has a lower computational cost, its resulting estimate is frequently far from the global optimum obtained by the global method.



— A —



— B —



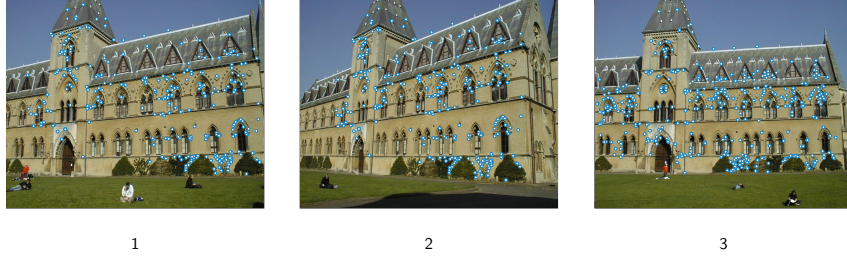
— C —



— D —

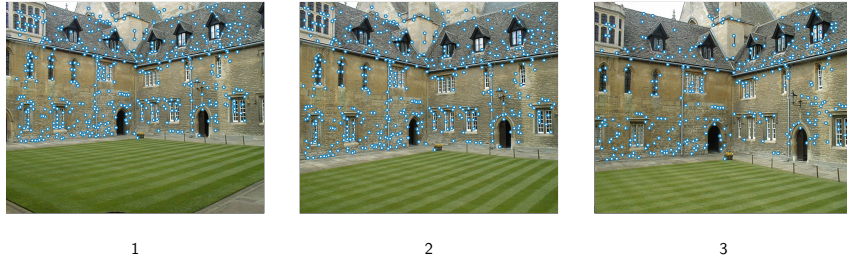
Views		Epipoles		$e_{\text{Init}}(\mathbf{F})$		$e_{\text{BA}}(\mathbf{F})$		Iter(\mathbf{F})		Time (s)	
		\mathbf{e}	\mathbf{e}'	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}
A	B	∞	∞	0.597617	0.65270	0.00252	0.00252	6	6	0.017	2.39
A	C	∞	∞	5.61506	5.61996	2.48258	0.00342	122	175	0.017	1.98
A	D	∞	∞	21.0855	21.5848	4.74837	0.00344	105	30	0.017	2.12
B	C	∞	∞	2.49098	1.91136	0.00260	0.00260	17	12	0.018	1.97
B	D	∞	∞	22.0071	23.6253	0.00268	0.00268	122	81	0.018	1.92
C	D	∞	∞	28.6586	28.6174	16.6507	0.25921	39	1001	0.018	2.1

Figure 6: Reprojection Error before ($e_{\text{Init}}(\mathbf{F})$) and after Bundle Adjustment ($e_{\text{BA}}(\mathbf{F})$), Number of Iterations (Iter(\mathbf{F})), and CPU time to compute \mathbf{F} (Time), obtained when combining pairs of images to obtain epipoles close to the images or toward infinity



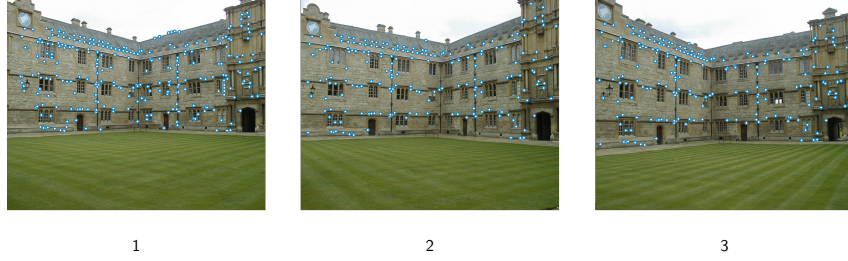
Views		nb Points	$e_{\text{Init}}(\mathbf{F})$		$e_{\text{BA}}(\mathbf{F})$		Iter(\mathbf{F})		Time (s)	
			\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}
1	2	310	18.27449	18.32095	11.39809	11.287591	65	57	0.035	2.05
2	3	439	61.84817	63.79435	41.81854	40.2224	29	28	0.020	2.22
1	3	439	42.67438	42.23867	28.65971	27.90128	40	32	0.043	2.10

Figure 7: Reprojection Error before ($e_{\text{Init}}(\mathbf{F})$) and after Bundle Adjustment ($e_{\text{BA}}(\mathbf{F})$), Number of Iterations (Iter(\mathbf{F})), and CPU time to compute \mathbf{F} (Time), obtained when combining pairs of images of the *Library* series



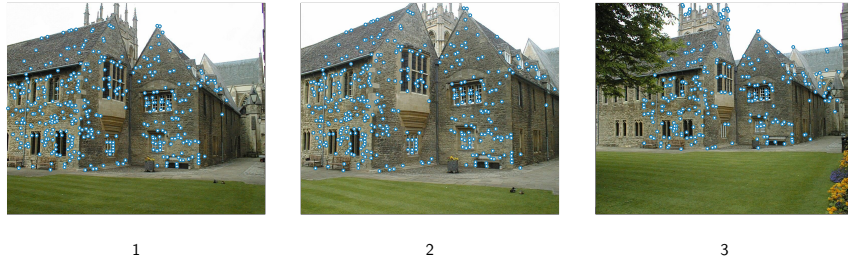
Views		nb Points	$e_{\text{Init}}(\mathbf{F})$		$e_{\text{BA}}(\mathbf{F})$		Iter(\mathbf{F})		Time (s)	
			\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}
1	2	485	15.1771	15.3126	9.8859	9.7360	77	55	0.048	2.02
2	3	439	61.84817	63.79435	41.81854	40.2224	29	28	0.020	2.22
1	3	384	51.3128	2.3690	7.5245	0.56180	5	52	0.040	2.71

Figure 8: Reprojection Error before ($e_{\text{Init}}(\mathbf{F})$) and after Bundle Adjustment ($e_{\text{BA}}(\mathbf{F})$), Number of Iterations (Iter(\mathbf{F})), and CPU time to compute \mathbf{F} (Time), obtained when combining pairs of images of the *Merton1* series



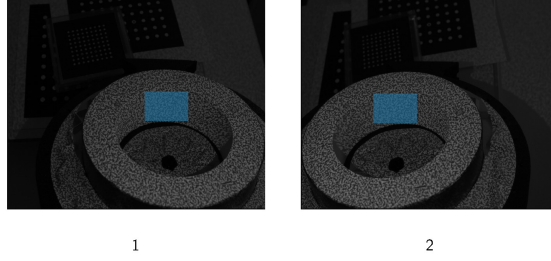
Views		nb Points	$e_{\text{Init}}(\mathbf{F})$		$e_{\text{BA}}(\mathbf{F})$		Iter(\mathbf{F})		Time (s)	
			\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}
1	2	345	24.7158	29.6530	14.0481	2.7599	69	42	0.037	2.91
2	3	197	143.431	154.880	73.8776	72.8977	14	19	0.026	2.52
1	3	270	59.8109	77.9734	30.3824	15.6367	38	32	0.031	2.37

Figure 9: Reprojection Error before ($e_{\text{Init}}(\mathbf{F})$) and after Bundle Adjustment ($e_{\text{BA}}(\mathbf{F})$), Number of Iterations (Iter(\mathbf{F})), and CPU time to compute \mathbf{F} (Time), obtained when combining pairs of images of the *Merton2* series



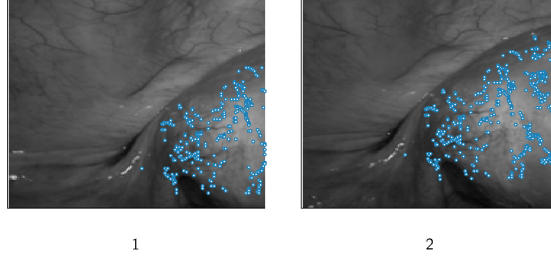
Views		nb Points	$e_{\text{Init}}(\mathbf{F})$		$e_{\text{BA}}(\mathbf{F})$		Iter(\mathbf{F})		Time (s)	
			\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}
1	2	400	62.7290	68.6030	52.6557	23.1740	5	32	0.041	3.03
2	3	197	135.950	140.801	83.3365	76.8614	13	14	0.025	2.52
1	3	264	116.7659	118.1754	24.7184	13.5496	9	11	0.032	2.39

Figure 10: Reprojection Error before ($e_{\text{Init}}(\mathbf{F})$) and after Bundle Adjustment ($e_{\text{BA}}(\mathbf{F})$), Number of Iterations (Iter(\mathbf{F})), and CPU time to compute \mathbf{F} (Time), obtained when combining pairs of images of the *Merton3* series



Views		nb Points	$e_{\text{Init}}(\mathbf{F})$		$e_{\text{BA}}(\mathbf{F})$		Iter(\mathbf{F})		Time (s)	
			\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}
1	2	6609	6.3142	6.4021	1.4701	0.2531	401	237	1.23	2.33

Figure 11: Reprojection Error before ($e_{\text{Init}}(\mathbf{F})$) and after Bundle Adjustment ($e_{\text{BA}}(\mathbf{F})$), Number of Iterations (Iter(\mathbf{F})), and CPU time to compute \mathbf{F} (Time), obtained when combining pairs of images of the *Cylinder* series. The matched points are located in blue bounding boxes.



Views		nb Points	$e_{\text{Init}}(\mathbf{F})$		$e_{\text{BA}}(\mathbf{F})$		Iter(\mathbf{F})		Time (s)	
			\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}	\mathbf{F}_{8Pt}	\mathbf{F}_{Gp}
1	2	730	5.6624	5.6366	3.8566	0.9386	163	401	0.07	1.69

Figure 12: Reprojection Error before ($e_{\text{Init}}(\mathbf{F})$) and after Bundle Adjustment ($e_{\text{BA}}(\mathbf{F})$), Number of Iterations (Iter(\mathbf{F})), and CPU time to compute \mathbf{F} (Time), obtained when combining pairs of images of the *Endoscope* series

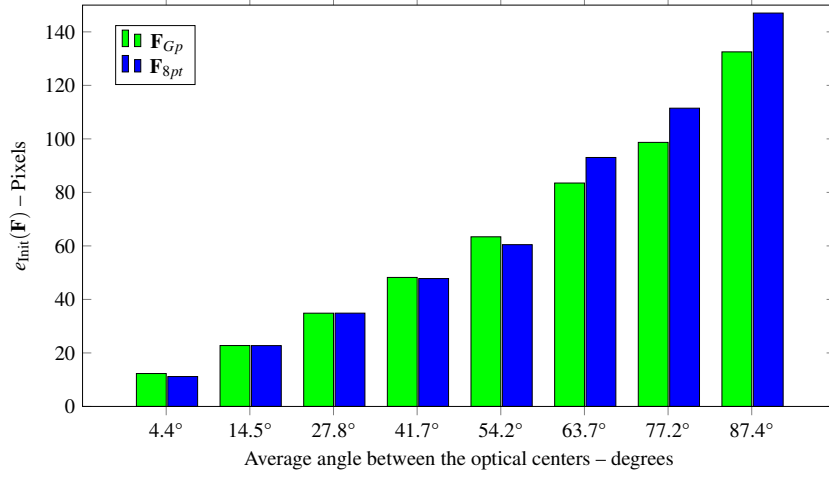


Figure 13: Initial re-projections errors measured against movement amplitude for the *dinosaur* series

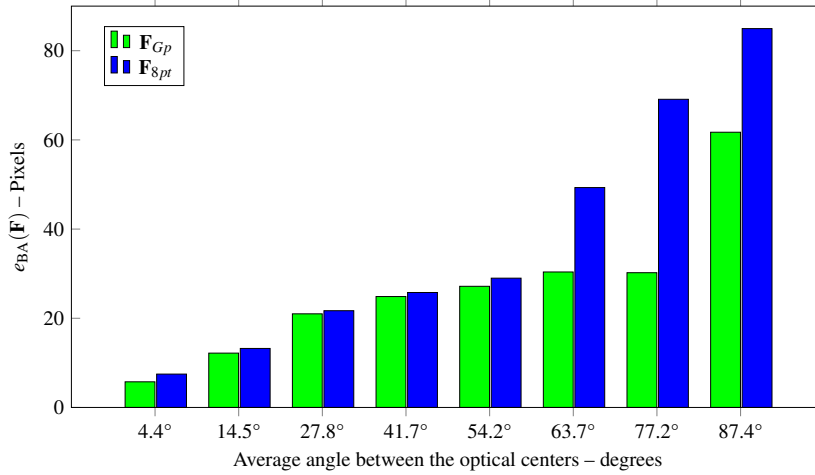


Figure 14: Final re-projections errors measured against movement amplitude for the *dinosaur* series

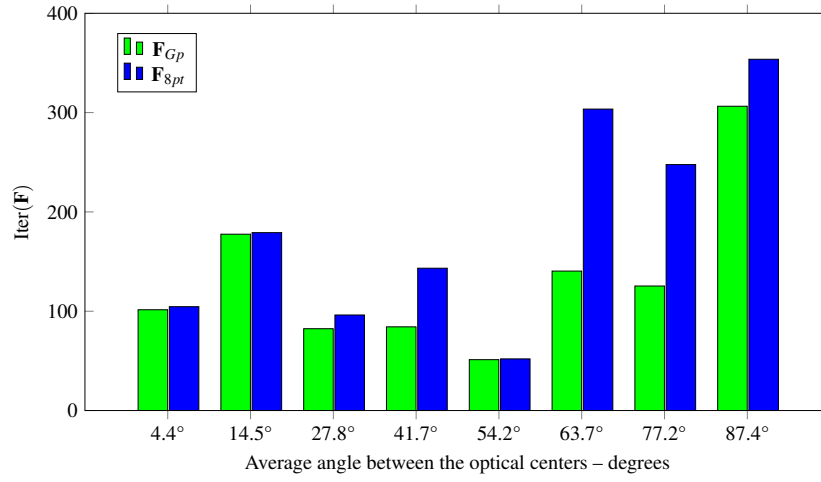


Figure 15: Number of iterations performed by Bundle-Adjustment to converge measured against movement amplitude for the *dinosaur* series

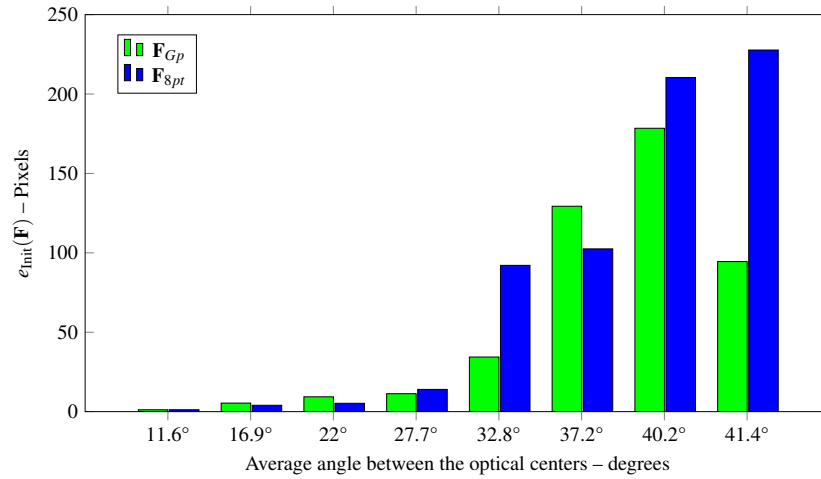


Figure 16: Initial re-projections errors measured against movement amplitude for the *House* series

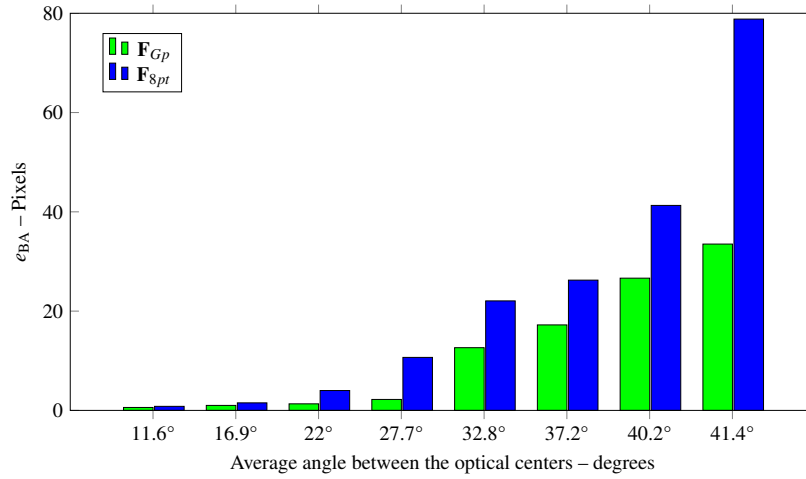


Figure 17: Final re-projections errors measured against movement amplitude for the *House* series

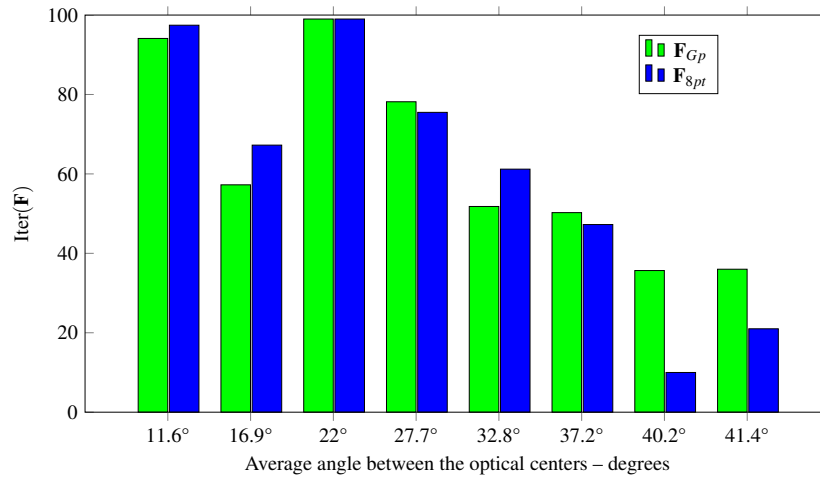


Figure 18: Number of iterations performed by Bundle-Adjustment to converge measured against movement amplitude for the *House* series

References

- [1] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, Sep 1981.
- [2] X. Armangué and J. Salvi. Overall view regarding fundamental matrix estimation. *Image and Vision Computing*, 21:205–220, 2003.
- [3] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 6:13–26, 1984.
- [4] R. Hartley. In Defence of the 8-point Algorithm. In *5th International Conference on Computer Vision (ICCV’95)*, pages 1064–1070, Boston (MA, USA), Jun 1995.
- [5] P. H. S. Torr and A. W. Fitzgibbon. Invariant fitting of two view geometry or ”in defiance of the 8 point algorithm”. 2002.
- [6] W. Chojnacki, M. J. Brooks, and A. Van Den Hengel. Revisiting hartley’s normalized eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1172–1177, 2003.
- [7] A. Bartoli and P. Sturm. Nonlinear estimation of the fundamental matrix with minimal parameters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(3):426–432, March 2004.
- [8] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11:796–817, 2001.
- [9] D. Henrion and J.B. Lasserre. Gloptipoly: Global optimization over polynomials with Matlab and SeDuMi. *Proceedings of IEEE Conference on Decision and Control*, Dec. 2002.
- [10] X. Xuelian. New fundamental matrix estimation method using global optimization. In *Proceedings of International Conference on Computer Application and System Modeling (ICCASM)*, Taiyuan, China, 22–24 October 2010.
- [11] G. Chesi, A. Garulli, A. Vicino, and R. Cipolla. Estimating the fundamental matrix via constrained least-squares: A convex approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:397–401, 2002.

- [12] F. Kahl and D. Henrion. Globally optimal estimates for geometric reconstruction problems. *International Journal of Computer Vision*, 74:3–15, 2007.
- [13] J. Salvi. *An Approach to Coded Structured Light to Obtain Three Dimensional Information*. PhD thesis, University of Girona, 1999.
- [14] P. Chen. Why not use the lm method for fundamental matrix estimation? *IET computer vision*, 4(4):286–294, 2010.
- [15] Q.-T. Luong and O. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43–76, 1996.
- [16] Z. Zhang. Determining the epipolar geometry and its uncertainty : A review. *International Journal of Computer Vision*, 27:161–195, 1998.
- [17] A. Van Den Hengel, W. Chojnacki, M. J. Brooks, and D. Gawley. A new constrained parameter estimator: Experiments in fundamental matrix computation. In *In Proceedings of the 13th British Machine Vision Conference*, September 2002.
- [18] W. Chojnacki, M. J. Brooks, A. Van Den Hengel, and D. Gawley. A new approach to constrained parameter estimation applicable to some computer vision problems. In *Statistical Methods in Video Processing Workshop held in conjunction with ECCV'02*, Copenhagen, Denmark, 2002.
- [19] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [20] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24:271–300, 1997.
- [21] F. Glover. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.*, 13(5):533–549, 1986.
- [22] P. Hansen and B. Jaumard. Algorithms for the maximum satisfiability problem. *Computing*, 44(4):279–303, 1990.
- [23] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28:497–520, 1960.
- [24] E. R. Hansen and G. W. Walster. *Global optimization using interval analysis*, 2nd Edn. Marcel Dekker, 2003.

- [25] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis*. SIAM, 2009.
- [26] T. Weise. *Global Optimization Algorithms - Theory and Application*. Thomas Weise, 2007-05-01 edition, 2007.
- [27] F. Kahl and D. Henrion. Globally optimal estimates for geometric reconstruction problems. In *Proc. IEEE Int. Conf. Computer Vision (ECCV)*, 2005.
- [28] Y. Zheng, S. Sugimoto, and M. Okutomi. A practical rank-constrained eight-point algorithm for fundamental matrix estimation. In *Computer Vision and Pattern Recognition, 2013. CVPR 2013. IEEE Conference on*, June 2013.
- [29] J. B. Lasserre. Optimisation globale et théorie des moments. *Comptes Rendus de l'Académie des Sciences*, 331:929–934, 2000.
- [30] J. B. Lasserre. *Moments, positive polynomials and their applications*, volume 1 of *Imperial College Press Optimization Series*. Imperial College Press, 2010.
- [31] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging Applications of Algebraic Geometry. IMA Volumes in Mathematics and its Applications*, 149:157–270, 2009.
- [32] B. Borchers. Csdp, a c library for semidefinite programming. *Optimization Methods and Software*, 11:613–623, 1999.
- [33] S. J. Benson and Y. Ye. DSDP5 user guide — software for semidefinite programming. Technical Report ANL/MCS-TM-277, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, September 2005. <http://www.mcs.anl.gov/~benson/dsdp>.
- [34] K. Fujisawa, M. Fukuda, M. Kojima, K. Nakata, M. Nakata, M. Yamashita, K. Fujisawa, M. Fukuda, and K. Kobayashi. Sdpa (semidefinite programming algorithm) – user’s manual. Technical report, Dept. Math. and Comp. Sciences – Tokyo Institute of Technology, 2003.
- [35] K.C. Toh, M.J. Todd, and R.H. Tutuncu. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical Programming*, 95:189–217, 2003.
- [36] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999. Version 1.05 available from <http://fewcal.kub.nl/sturm>.

- [37] J. Nie. Optimality conditions and finite convergence of Lasserre's hierarchy. Technical report, Dept. Math. – Univ. of California at San Diego, 2012.
- [38] D. Henrion, J.B Lasserre, and J. Löfberg. Gloptipoly 3: moments, optimization and semidefinite programming. *Optim. Methods and Software*, 24:761–779, 2009.
- [39] J. Lfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- [40] K.C. Toh, M.J. Todd, and R.H. Tutuncu. Sdpt3 — a matlab software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.
- [41] Q.-T. Luong and T. Viéville. Canonical representations for the geometries of multiple projective views. *Comput. Vis. Image Underst.*, 64:193–229, September 1996.
- [42] R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.